



max planck institut
informatik

Computational Thinking

Maschinelles Lernen

Kurt Mehlhorn

Kosta Panagioutou

Max-Planck-Institut für Informatik

Übersicht

- Lernen: Begriff, typische Aufgaben, Arten von Lernen
- Spamerkennung
- Handschriftenerkennung
 - mit und ohne Trainingsdaten
- Weitere Beispiele
 - Personenerkennung
 - Autonome Roboter

Lernen

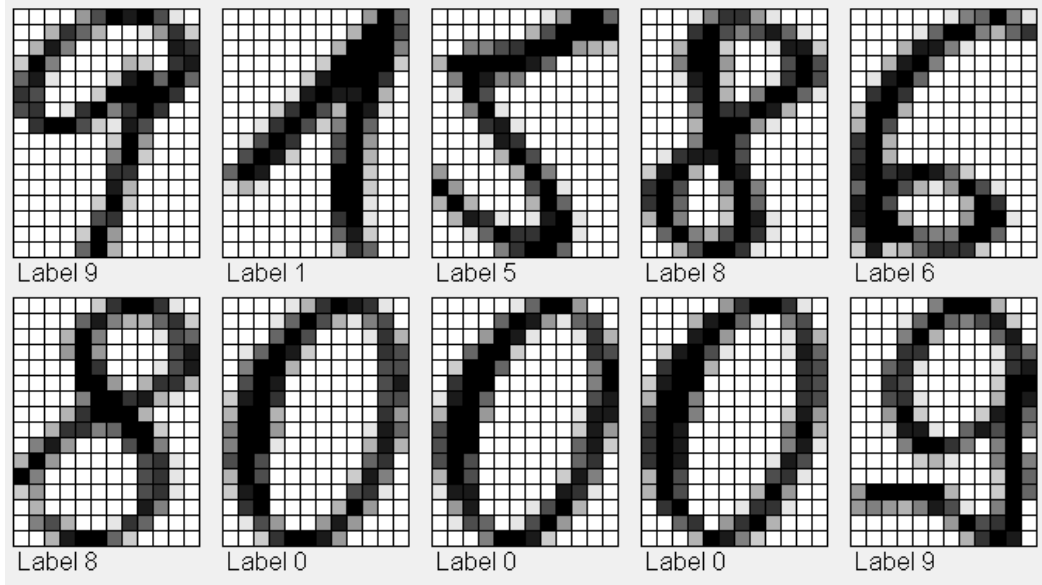
- Lernen aus einer Erfahrung bezüglich einer Menge von Aufgaben und einem Performanzmaß bedeutet besseres Verhalten
- Fähigkeit, Verhalten zu verbessern auf Grund von Erfahrungen
- Verallgemeinern von Erfahrungen
- Programmieren durch Beispiele

Typische Aufgaben

- Klassifikation: Spam versus Ham, Ziffernerkennung
- Objekterkennung: identifiziere Personen, Autos, ... auf Fotos und in Videos
- Robotersteuerung, lerne Autofahren, lerne eine Landkarte
- Spracherkennung, Übersetzung, Sprachverstehen

Arten von Lernen

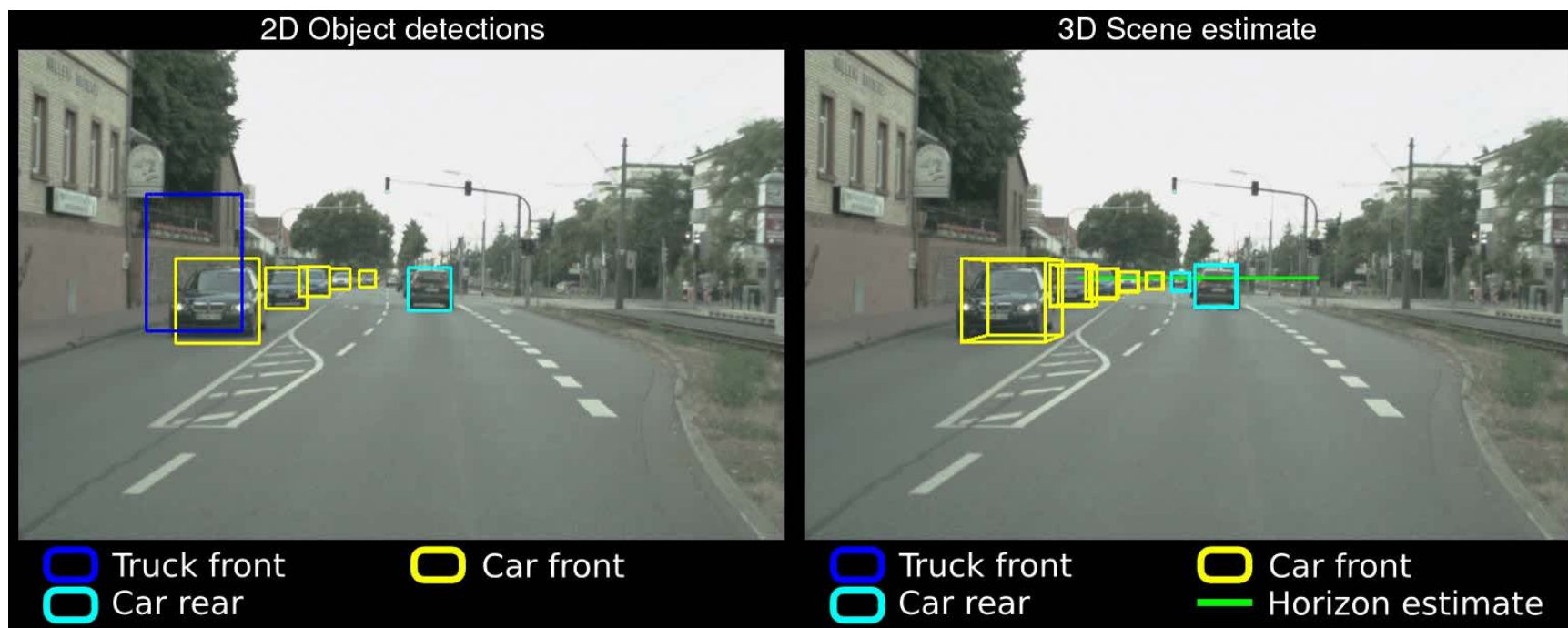
- Supervised: mit Trainingsdaten oder sogar mit Lehrer



- Unsupervised: ohne Trainingsdaten; dann ist es mehr Entdecken als Lernen

Objekterkennung

Abteilung Schiele: MPI für Informatik



Personenerkennung

Abteilung Schiele: MPI für Informatik



Bayessche Regel

(englischer Pfarrer und Mathematiker, 17??)

In einem Sack sind 900 Äpfel und 100 Paprika. Von den Äpfeln sind 10% rot und 90% grün. Bei den Paprika sind es jeweils 50%.

Ich entnehme eine Frucht zufällig. Sie ist rot. Was für eine Frucht ist es?

- Bayes: entscheide dich für den wahrscheinlicheren Fall.

Bayessche Regel

(englischer Pfarrer und Mathematiker, 17??)

In einem Sack sind 900 Äpfel und 100 Paprika. Von den Äpfeln sind 10% rot und 90% grün. Bei den Paprika sind es jeweils 50%.

$$P(\text{Apfel} \mid \text{rot}) = P(\text{Apfel}) * P(\text{rot} \mid \text{Apfel}) / P(\text{rot})$$

$P(\text{Apfel} \mid \text{rot}) * P(\text{rot})$ und $P(\text{rot} \mid \text{Apfel}) * P(\text{Apfel})$ sind beide gleich $P(\text{Apfel und rot})$

Spam versus Ham (Junk Mail)

- Absenderbasiert
 - email von Bekannten ist kein Spam
 - Schwarze Listen
- Inhaltsbasiert
 - Nutzer klassifiziert emails als gut und schlecht; System lernt daraus; Nutzer muss immer weniger eingreifen

Inhaltsbasierte Filter

- In der Trainingsphase lernen wir
 - Wahrscheinlichkeit von Ham und Spam
 - Jeweils Wahrscheinlichkeiten für Worte

- Ham

Freund	Vorlesung	Algorithmus	Geld	Viagra	schnell
0.1	0.3	0.3	0.1	0.1	0.1

- Spam

Freund	Vorlesung	Algorithmus	Geld	Viagra	schnell
0.2	0.1	0.1	0.2	0.3	0.1

Trainingsphase

- Nutzer klassifiziert emails als Spam und Ham (damit beide Wahrscheinlichkeiten)
- Sei n die Gesamtlänge meiner guten emails (in Worten), sei v die Anzahl der Vorkommen eines bestimmten Wortes
- Wahrscheinlichkeit des Wortes in Ham
 $= v/n$

Inhaltsbasierte Filter (Bayes Modell)

- Wahrscheinlichkeitsverteilung auf Worten

- Ham

Freund	Vorlesung	Algorithmus	Geld	Viagra	schnell
0.1	0.3	0.3	0.1	0.1	0.1

- Spam

Freund	Vorlesung	Algorithmus	Geld	Viagra	schnell
0.2	0.1	0.1	0.2	0.3	0.1

- $P(\text{Text} | \text{Ham}) = \text{Produkt der Wahrscheinlichkeiten der Worte im Text}$

Inhaltsbasierte Filter

- Ham

Freund	Vorlesung	Algorithmus	Geld	Viagra	schnell
0.1	0.3	0.3	0.1	0.1	0.1

- Spam

Freund	Vorlesung	Algorithmus	Geld	Viagra	schnell
0.2	0.1	0.1	0.2	0.3	0.1

- Viagra Geld Freund

– Falls Ham: $0.1 \times 0.1 \times 0.1 = 1/1000$

– Falls Spam: $0.3 \times 0.2 \times 0.2 = 12/1000$ also Spam

Inhaltsbasierte Filter

- Ham

Freund	Vorlesung	Algorithmus	Geld	Viagra	schnell
0.1	0.3	0.3	0.1	0.1	0.1

- Spam

Freund	Vorlesung	Algorithmus	Geld	Viagra	schnell
0.2	0.1	0.1	0.2	0.3	0.1

- Vorlesung Algorithmus schnell
 - Falls Ham: $0.3 \times 0.3 \times 0.1 = 9/1000$
 - Falls Spam: $0.1 \times 0.1 \times 0.1 = 1/1000$ also Ham

Inhaltsbasierte Filter

- Ham

Freund	Vorlesung	Algorithmus	Geld	Viagra	schnell
0.1	0.3	0.3	0.1	0.1	0.1

- Spam

Freund	Vorlesung	Algorithmus	Geld	Viagra	schnell
0.2	0.1	0.1	0.2	0.3	0.1

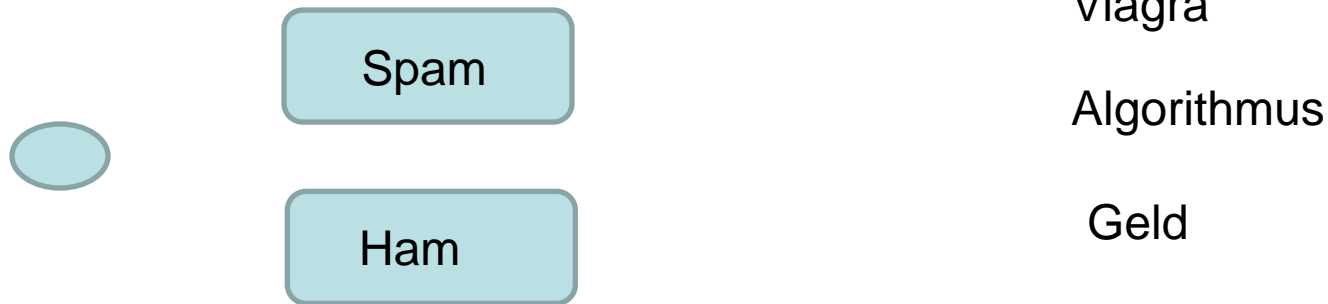
- Viagra Algorithmus schnell
 - Falls Ham: $0.1 \times 0.3 \times 0.1 = 3/1000$
 - Falls Spam: $0.3 \times 0.1 \times 0.1 = 3/1000$ also ???

Nutzungsphase

- Nutzungsphase: System klassifiziert
- Verteilung wird weiter trainiert (seltene Worte)
- Nutzer kann widersprechen
- Spammer lernen auch dazu: V!agra statt Viagra

Zusammenfassung

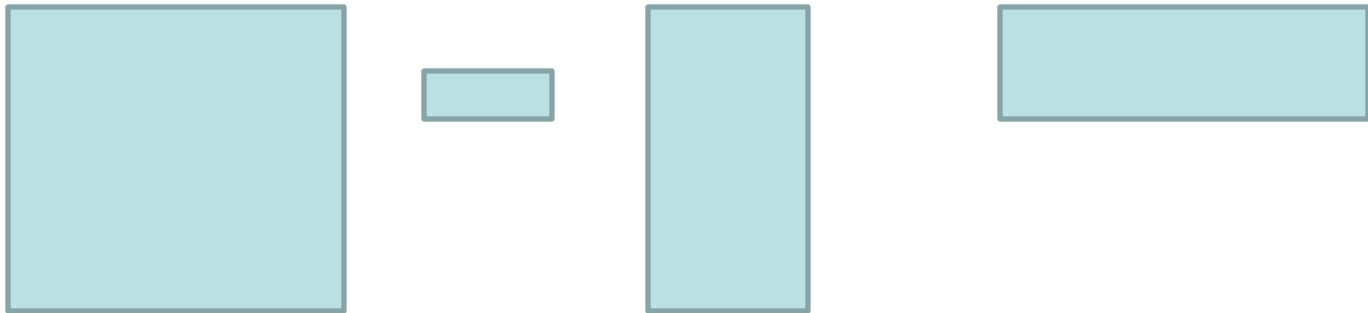
- Wir haben Modell, wie Ereignisse (emails) erzeugt werden



- Lernen das Modell in der Trainingsphase
- Geben für jedes Ereignis die wahrscheinlichste Erklärung (Bayes)
- Klassifizierung in: Geschäftspost, Privatpost, Spam

Verborgene Teile des Modells

- Daten = Worthäufigkeiten in Dokumenten
- Es gibt Themen etwa Sport, Politik, ...
- Es gibt Worthäufigkeiten für Topics
- Dokumente haben ein Themenprofil
- Finde Faktorisierung



Ziffernerkennung Übersicht

- Trainingsdaten: handgeschriebene Ziffern mit Label (die Ziffer)
- Ziffern als Vektoren
- Abstandsbegriffe
- Erkennungsverfahren: nächster Nachbar, k-nächste Nachbarn, typische Buchstaben durch Mittelwertbildung
- Und ohne Training: k-means

Trainingsdaten I



Grundidee

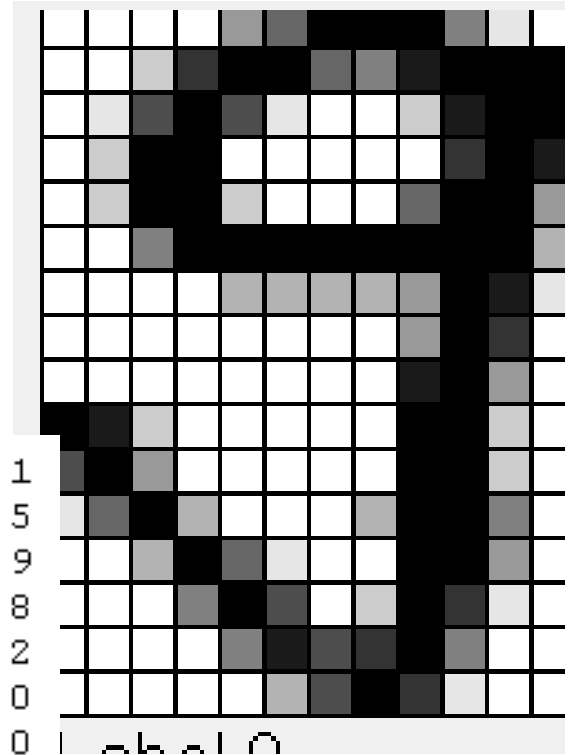
- Zwei Bilder repräsentieren die gleiche Ziffer, wenn die Bilder sich ähnlich sind
- Ähnlich = ähnliche Grauwertverteilung
- Wir stellen Bilder als eine Matrix von Grauwerten dar

Bilder = Matrizen von Zahlen

Ziffer = 12 x 16 Matrix
von Grauwerten in [0,1]

Vektor von Grauwerten
der Länge 192

0.0	0.0	0.0	0.2	0.3	0.4	0.8	1.0	1.0	0.7	0.3	0.1
0.0	0.5	0.8	0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.5
0.3	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.9
0.3	1.0	1.0	1.0	1.0	1.0	0.8	1.0	1.0	1.0	1.0	0.8
0.3	1.0	1.0	1.0	1.0	0.8	0.1	0.9	1.0	1.0	0.8	0.2
0.0	0.7	1.0	1.0	1.0	0.8	0.8	1.0	1.0	1.0	0.4	0.0
0.0	0.2	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.3	0.0	0.0
0.0	0.1	0.7	1.0	1.0	1.0	1.0	0.8	0.1	0.0	0.0	0.0
0.0	0.6	1.0	1.0	1.0	1.0	0.9	0.1	0.0	0.0	0.0	0.0
0.6	1.0	1.0	1.0	1.0	1.0	1.0	0.3	0.0	0.0	0.0	0.0
0.8	1.0	1.0	0.5	0.1	0.7	1.0	0.8	0.2	0.0	0.0	0.0
0.5	1.0	1.0	0.3	0.0	0.0	0.9	1.0	0.9	0.1	0.0	0.0
0.4	1.0	1.0	0.3	0.0	0.0	0.5	1.0	1.0	0.5	0.0	0.0
0.0	0.4	1.0	1.0	0.5	0.3	0.5	1.0	1.0	1.0	0.2	0.0
0.0	0.0	0.5	1.0	1.0	1.0	1.0	1.0	1.0	0.8	0.1	0.0
0.0	0.0	0.0	0.2	0.5	0.7	1.0	1.0	0.9	0.3	0.0	0.0



Ähnlichkeit von Vektoren allgemein

- Zwei Vektoren x und y sind ähnlich,
 - wenn $x - y$ kurz ist
 - Wenn der aufgespannte Winkel klein ist

Länge eines Vektors $x = (x_1, \dots, x_n)$

$$\|x\| = \sqrt{x_1^2 + x_2^2 + \dots + x_n^2}$$

Winkel zwischen x und y

$$\cos \alpha = x \cdot y / \|x\| \cdot \|y\|$$

Ähnlichkeit von Vektoren

- Zwei Vektoren x und y sind ähnlich,
 - wenn $x - y$ kurz ist
 - Wenn der aufgespannte Winkel klein ist
- Wahl des richtigen Ähnlichkeitsbegriffs ist eine schwarze Kunst

Verfahren: Nearest Neighbor

um die Bedeutung des Bildes p zu finden, finde das Trainingsbild x mit $\text{dist}(p,x)$ minimal (durch lineare Suche über alle Trainingsdaten)

Gib das Label von x aus

- Erkennungsrate 0.934
- Majority of 3 nearest neighbors 0.945
- Majority of 9 nearest neighbors 0.920
- Mit cos-Distanz 0.940
- Majority of 3 nearest neighbors 0.920

Detaillierte Ergebnisse

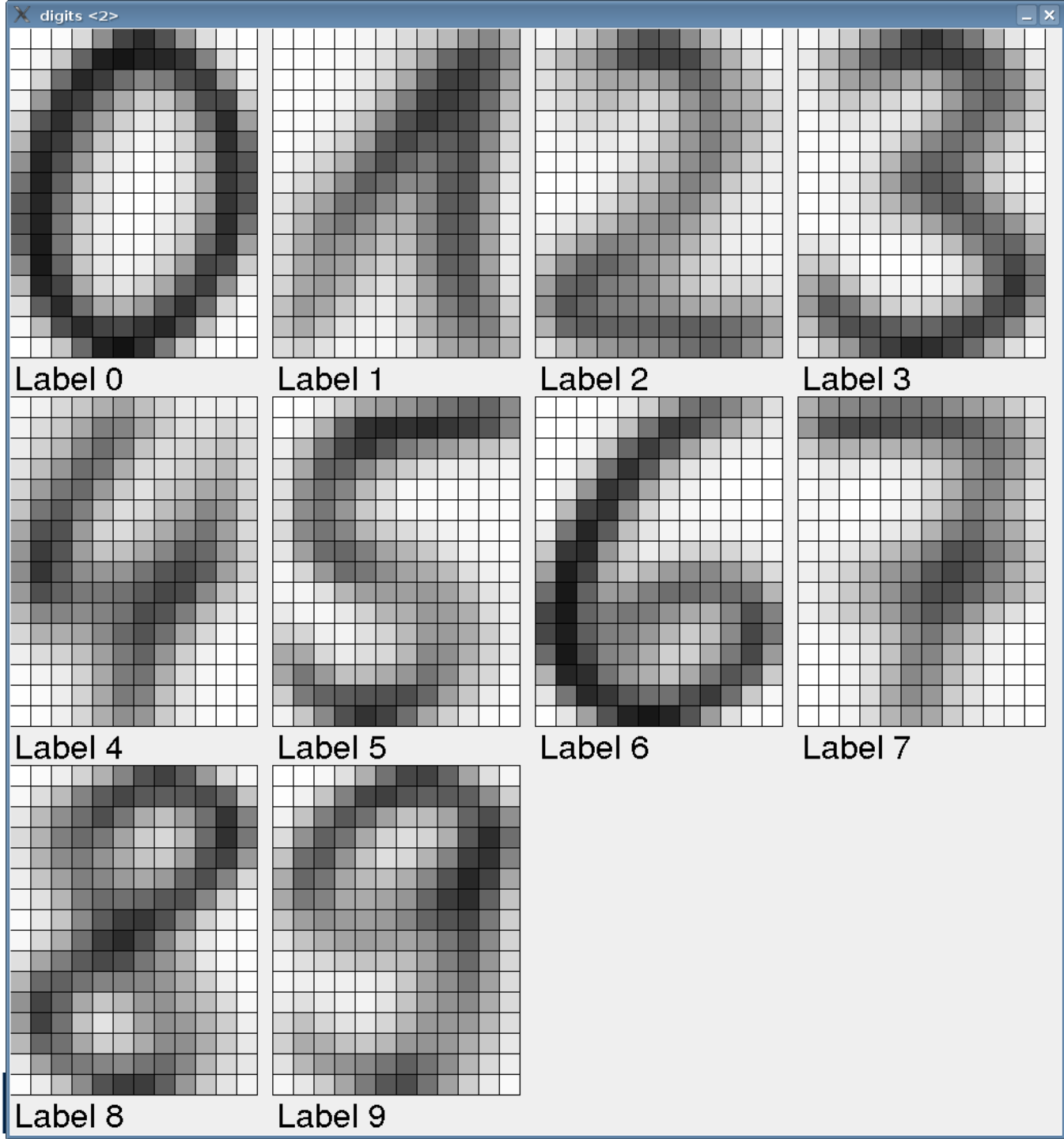
digit = 0 accuracy = 1.0
digit = 1 accuracy = 0.90
digit = 2 accuracy = 0.92
digit = 3 accuracy = 1.0
digit = 4 accuracy = 0.95
digit = 5 accuracy = 0.85
digit = 6 accuracy = 0.84
digit = 7 accuracy = 1.0
digit = 8 accuracy = 0.7
digit = 9 accuracy = 0.94

Klassifizierung dauert lang, da
jedes Mal ALLE Trainingsdaten
angeschaut werden

digit = 0 accuracy = 1.0
digit = 1 accuracy = 0.95
digit = 2 accuracy = 0.92
digit = 3 accuracy = 1.0
digit = 4 accuracy = 1.0
digit = 5 accuracy = 0.78
digit = 6 accuracy = 0.77
digit = 7 accuracy = 0.95
digit = 8 accuracy = 0.8
digit = 9 accuracy = 1.0

Klassen → Klassenzentren

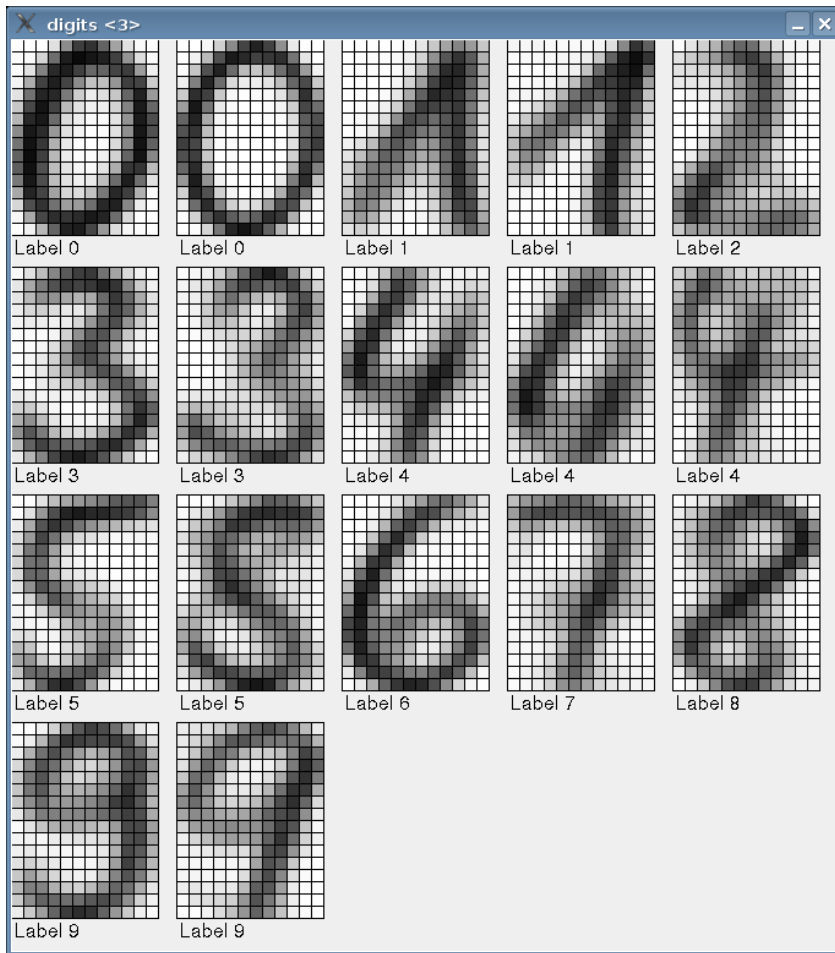
- Vorerechne für jede Klasse (Ziffer) das Klassenzentrum durch Durchschnittsbildung
- Suche: finde das nächstgelegene Zentrum (10 Vergleiche)
- Erkennungsrate: 0.854
- Mit cos-distance 0.894



Lernen ohne Training

- Vorbereiten der Trainingsdaten ist mühsam
- Können wir Klassen entdecken, ohne das uns Klassenlabels gesagt werden?
- Automatische Klassifizierung durch k-means Algorithmus.
- Danach Vergleich mit den Klassenzentren
- $k=10$, Rate 0.683
- $k=17$, 0.733
- with cos-distance,
- $k=10$, 0.728
- $k=17$, 0.783

K-means Algorithmus



- Automatische Klassifizierung in 17 Klassen
- Danach (!!!) Zuweisung eines Labels per Hand und Wegwerfen von schlechten Zentren
- Identifiziert die zwei Schreibweisen der Neun

k-means Algorithmus

Versucht Vektoren in k sinnvolle Cluster (Haufen) einzuteilen.

1. Starte mit k beliebigen Zentren
2. Weise jeden Punkt dem nächstgelegenen Zentrum zu und bilde so k Cluster
3. Berechne für jeden Cluster seinen Schwerpunkt; das sind die neuen Zentren
4. Gehe nach 2.

Personenerkennung

